A Hybrid Emulation Environment for Airborne Wireless Networks

Marco Carvalho, Adrián Granados, Carlos Pérez, Marco Arguedas,
Michael Muccio, Joseph Suprenant, Daniel Hague, and Brendon Poland

Abstract. Airborne networks are among some of the most challenging communication environments for protocol design and validation. Three dimensional and highly dynamic in nature, airborne network links are generally difficult to simulate accurately and often expensive to recreate in field experiments. The physical characteristics of the data links in these kinds of networks are not only influenced by the usual environmental factors and interferences, but also by the frequent changes in the relative angles and positions between antennas, shadowing effects of the airframe, non-uniform ground effects and other factors. In this work we summarize our research for the development of mLab, a hybrid emulation environment for airborne networks that combines theoretical path loss models with statistical data-driven models to create a low-cost, high fidelity, specialized emulation environment for the development, evaluation, and validation of protocols and algorithms for airborne wireless networks.

1 Introduction

The majority of the mobile ad hoc network (MANET) research is based on computer simulations, emulation network environments and field tests. While some important

Marco Carvalho
Florida Institute of Technology, 150 W University Blvd., Melbourne, FL 32901 USA, and
Institute for Human and Machine Cognition, 15 SE Osceola Ave., Ocala, FL 34471, USA
e-mail: mcarvalho@fit.edu

Adrián Granados · Carlos Pérez · Marco Arguedas
Florida Institute for Human and Machine Cognition, 15 SE Osceola Ave.,
Ocala, FL 34471, USA
e-mail: {agranados,cperez,marguedas}@ihmc.us

Michael Muccio · Joseph Suprenant · Daniel Hague · Brendon Poland
U.S. Air Force Research Laboratory, 26 Electronic Parkway, Rome, NY, USA
e-mail: {michael.muccio,joseph.suprenant}@rl.af.mil,
{daniel.hague,brendon.poland}@rl.af.mil

A. Byrski et al. (Eds.): Advances in Intelligent Modelling and Simulation, SCI 416, pp. 111–129.
springerlink.com

theoretical work does exist in the literature, they generally constitute less than 25% of the body of work in field, as noted by Kurkowski et al. [12] in their review of five years of published research. Thus, wireless modeling and simulation play an important role in MANET, and airborne networks research.

In most wireless simulation environments, theoretical data link models such as Free Space or Two Ray Ground [15] are used to recreate the effects of the physical environment [6, 9]. The relative position of nodes, as well as their mobility patterns and physical interface settings, were normally treated as inputs to mathematical models used to calculate the quality of individual links. In most simulation tools and frameworks [1, 4, 7, 8, 2, 13] mathematical link models are then used to calculate the rates of drop or corrupt data-rates necessary to reflect specific scenarios and operational settings. However, the quality of the simulation is directly affected by the accuracy of such models [17], as well as the simulation and experimental approaches [11, 14, 12].

Not surprisingly, the alternative approach for validation and development is to rely on field experiments, which is usually costly, time consuming and very difficult to replicate. In the case of airborne networks the issue is even more complicated. Three dimensional and highly dynamic in nature, airborne network links are generally difficult to simulate accurately, and often expensive to recreate in field experiments. The physical characteristics of the data links in these kinds of networks are not only influenced by the usual environmental factors and interferences, but also by the frequent changes in the relative angles and positions between antennas, shadowing effects of the airframe, non-uniform ground effects and other factors.

From a protocol design perspective, an accurate emulation of these kinds of link-effects is important to enable the experimental validation of laboratory results, and to better understand the behavior of the protocol. However, popular non-commercial wireless network simulators [1, 3, 4, 10] rely solely on common theoretical radio propagation models that are often constrained to specific environments and communication parameters, lacking of models to emulate the environmental and external effects that influence the characteristics of wireless links.

In this context, empirical data-driven radio propagation models have been previously proposed [5, 16, 19] to create more reliable emulation environments for different target scenarios and varying wireless technologies. Hence, we have developed mLab, a hybrid emulation environment for airborne networks that combines theoretical path loss models with statistical data-driven models to create a low-cost, high fidelity, specialized emulation environment for the development, evaluation and validation of protocols and algorithms for airborne wireless networks.

In the following sections, we present the design and architecture of mLab, and also discuss the integration of mLab and EMANE (the Extendable Mobile Ad-hoc Network Emulator) to provide additional link enforcement mechanisms as well as to facilitate the reuse and sharing of mLab's statistical link models. Finally, we give a detailed description of our approach for generating data-driven corrective models using the datasets from the Protocol Emulation for Next Generation Wireless UAS Networks (PENGWUN) project, an internal effort of the Air Force Research

Laboratory (AFRL). We conclude this chapter, by presenting the experimental results of the tests and evaluation of our proposed statistical data-driven models.

2 The mLab Hybrid Emulation Testbed

The mLab hybrid emulation testbed is one of the components of the PENGWUN project. The architecture of the mLab (Fig. 1) testbed has four components: the hardware infrastructure, the modeling component, the coordination component, and the enforcement component. In particular, the last three components are responsible, respectively, for the modeling of wireless links and node behaviors, the estimation of the communications topology and the characteristics of each communication link, and the enforcement of policies (at the lowest possible levels) for wireless link emulation.

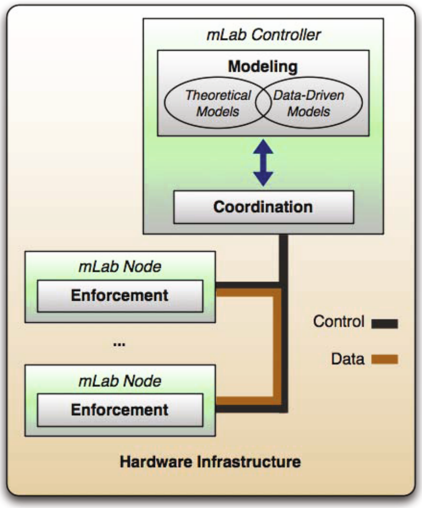


Fig. 1 Conceptual mLab architecture with enforcement components.

The emulation environment is recreated on top of a hardware infrastructure that must provide the necessary resources for the emulation, model enforcement and system coordination. There are several approaches for creating the hardware testbed and most often, deployments tend to include fixed infrastructures with sets of nodes dedicated to the emulation itself, and a few nodes dedicated to system configuration and control. Our initial design for mLab relied on a fixed set of nodes connected through two independent networks (one for data and one for control), with one platform node (the controller node) acting as the modeling and coordination element. In this configuration the user interfaces, and all other emulation nodes may be directly or remotely connected to the controller. The only requirement is that network latency at the physical level must be significantly smaller than the required emulated latencies, and the capacity at the physical level must be significantly larger than the capacity of the emulated system.

Testbed nodes have two interfaces, one connected to a data-only high bandwidth switched network and the other one connected to a control network, which is used for the control and monitoring of link behavior and network policies. The current version of mLab replaces the physical infrastructure by a virtualized environment where the controller can be dynamically connected with emulation nodes running as virtualized machines in a cloud environment. This new design for the physical infrastructure is functionally similar to the original concept.

The modeling component is responsible for modeling the physical links. As a modular component, it can be replaced to represent any arbitrary physical link using theoretical propagation models, data-driven statistical models, or a combination of them. The coordination component holds a model of the network topology represented as a graph structure. Each vertex in the graph maps to a specific host in the testbed. The coordination component continuously monitors the position and other attributes of the nodes to re-calculate and parameterize the models and enforcement components when network conditions change.

As an example, when the enforcement is performed by virtual drivers at each emulation node, the characteristics that define a communications link are only two, (1) a probability distribution for packet loss and (2) a probability distribution for packet delay. For each link in the graph, the coordination component maintains four distributions, two (delay and packet loss) for each direction. As nodes move, and traffic pattern changes, the distributions are then adjusted based on the physical link models. The parameters of each distribution are then used to configure the actual data links between nodes.

mLab uses a centralized approach designed to support the concurrent execution of multiple experiments. A *controller* node encapsulates the modeling and coordination components, maintaining the global view of the network topology for each active experiment, and calculating the physical link characteristics. Whenever a node shuts-down the vertex is removed from the graph. Upon startup, each node registers with the controller and a vertex is then created in the graph with the unique identifier of the node.

The enforcement component is a distributed software component that exists both at the controller node and at each host in the testbed. It provides coordination of policies related to current link conditions and communication constraints, and provides the enforcement capabilities (at the lowest possible levels) of packet loss and delays for the emulation. In our current implementation, the enforcement infrastructure is provided through customized virtual network drivers, directly interfacing with the controller node, or through EMANE, which provides pluggable Medium Access Control (MAC) and physical (PHY) layers for emulation of heterogeneous mobile ad-hoc networks.

3 mLab—EMANE Integration

EMANE is an infrastructure used for emulation of simple as well as complex heterogeneous mobile ad-hoc networks [13]. It supports pluggable MAC and PHY layers that allow for the emulation of commercial and tactical networks with multiple tiers and varying wireless technologies. It also supports multiple platforms (Windows, Linux and Mac OS X) and provides mechanisms to set up small as well as large-scale emulations using centralized, distributed or virtualized deployments. EMANE provides a modular architecture with well-defined APIs to allow independent development of emulation functionality for different radio models (network emulation modules), boundary interfaces between emulation and applications (transports), and distribution of emulation environmental data (events).

In EMANE, each node is represented by an instance of an emulation stack, which is comprised of three components: transport, Network Emulation Module (NEM), and Over-the-Air (OTA) Manager. This set of components encapsulate the functionality necessary to transmit, receive, and operate data routed through the emulation space. In particular, the transport component is responsible for handling data to and from the emulation space and interfaces with the underlying operating system using different platform-dependent mechanisms such as TUN/TAP (Linux, Mac OS X) and WinTap (Windows). On the other hand, a NEM provides emulation functionality for the MAC and PHY layers, including CSMA, TDMA, queue management, and adaptive transport protocols for the MAC layer, and waveform timing, half-duplex operations, interference modeling, probability of reception, out-of-band packets, and more for the PHY layer. Finally, the OTA Manager provides the necessary messaging infrastructure to deliver emulation radio model data to all nodes participating in the emulation. Control messages from the OTA Manager are distributed to each NEM through a multicast channel. Fig. 2 shows a diagram of the EMANE emulation stack.

The NEM and its corresponding MAC and PHY components are responsible for the enforcement of link characteristics. Hence, packets in the emulation space contain a special EMANE identification header and it is up to each layer to decide whether a packet is dropped or delayed before passing it to the upper layers. Each NEM requires a base configuration that specifies the network emulation model and the parameter values to configure the capabilities provided by the MAC and PHY components of the model. This configuration dictates the behavior of the emulation stack, which can be modified through the dissemination of emulation environmental data, such as location information or path loss, to emulate the dynamics of the network (e.g. node movement). This data can be distributed to NEMs in real time from the EMANE Event Service using a multicast channel, similarly to how the OTA Manager distributes control messages.

The type of environmental data that can be used to change communication constraints and how this data affects the behavior of the emulation depends on the configured network emulation model of the NEMs. For example, the Universal PHY Layer, a common PHY implementation for the various MAC layers that can be used with EMANE, may accept Location events specifying the latitude, longitude and

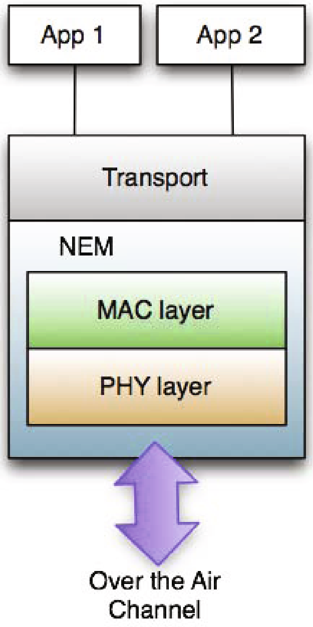


Fig. 2 EMANE emulation stack.

altitude of the NEM. When the Universal PHY Layer receives a Location event, the PHY implementation automatically computes the path loss using the most recent location information for all NEMs and the selected propagation model (Free Space or Two Ray Ground).

Alternatively, the Universal PHY Layer also supports path loss as a type of emulation environmental data that can be utilized to change link emulation conditions between NEMs. In this case, path loss computation is performed by a third-party outside of EMANE (e.g. mLab). When the Universal PHY Layer receives a *Pathloss* event containing a variable list of path loss and reverse path loss values between the receiving NEM and one or more NEMs in the emulation, the PHY implementation changes the characterization of the links to other NEMs based on the specified path loss values.

The integration between mLab and EMANE takes advantage of EMANE's network emulation models and event functionality to specify communication constraints and enforce link characteristics between mLab emulation nodes. In particular, EMANE's Universal PHY Layer's modes of operation (location or path loss) are used by mLab to perform link characterization using either EMANE's theoretical propagation models, mLab's theoretical propagation models, or the statistical data-driven models derived from the PENGWUN datasets that we have developed as part of this effort.

In mLab, emulation modes may be configured with multiple network interfaces over one or more different media (channels). In the case of EMANE, a medium corresponds to a network emulation model, such as IEEE80211abg, which emulates

IEEE 802.11 MAC layer’s Distributed Coordination Function (DCF) channel access scheme and IEEE 802.11 Direct Spread Spectrum Sequence (DSS) and Orthogonal Frequency Division Multiplexing (OFDM) signals in space, or RFPipe, a model that provides low fidelity emulation of a variety of waveforms.

In order to integrate EMANE’s capabilities into mLab, we had to provide mLab with the ability to automatically generate and deploy EMANE’s XML configuration files, as well as the ability to generate EMANE *Pathloss* and *Location* events. Additionally, we also extended EMANE with a simple feedback mechanism that allows mLab to receive notification for changes in path loss, and implemented a TxPower event that allows mLab to dynamically control the transmitter power settings of a NEM.

```
<?xml version=’1.0’ encoding=’utf-8’?>
<scenario name="MyScenarioDefinition"
          description="A small example scenario.">

  <!-- Defines a medium that uses EMANE’s IEEE 802.11abg model -->

  <medium id="ename80211abg">
    <enforcer classname="us.ihmc.mlab.emane.EmaneEnforcer">
      <model classname="us.ihmc.mlab.emane.Emane80211EnforcerModel">
        <param name="mac.rtsthresh" value="100"/>
        <param name="phy.txpower" value="15"/>
        <param name="phy.frequency" value="2417000"/>
        <param name="phy.pathlossmode" value="freespace"/>
      </model>
    </enforcer>
  </medium>

  <!-- Defines nodes -->

  <node id="1" label="UAV">
     <interface medium_id="emane80211abg"
                address="192.168.0.1"
                netmask="255.255.255.0" />
     <position lat="29.187617746963323"
               lon="-82.13927366751467"
               alt="400.0"
               roll="0.0"
               pitch="0.0"
               yaw="0.0"/>
  </node>

  <node id ="2" label="GN">
     <interface medium_id="emane80211abg"
                address="192.168.0.2"
                netmask="255.255.255.0"/>
     <position lat="29.18655204125978"
               lon="-82.13563653875342"
               alt="80.0"
               roll="0.0"
               pitch="0.0"
               yaw="0.0"/>
  </node>

</scenario>
```

Fig. 3 Example of mLab XML Scenario File

3.1 Generation and Deployment of EMANE's XML Configuration Files

When the mLab controller loads a scenario file (Fig. 3), the EMANE enforcer component in the controller generates the necessary EMANE configuration files for the physical nodes that will take part in the emulation. Then, these XML configuration files are deployed to create and configure a NEM with the given network emulation model (e.g. IEEE80211abg MAC and Universal PHY Layer) to emulate the desired medium. An XML scenario may contain one or more medium definitions, each of them with specific model configuration parameters. Each medium is given an ID, which is then used to associate the node's network interface with a particular network emulation model.

A subset of the most relevant configuration parameters of each EMANE's network emulation model can be specified through the definition of the medium in the mLab scenario file. Parameter names that start with the prefix "phy." map directly to the configuration parameters of EMANE's Universal PHY Layer, which is used by both the IEEE80211 and RFPipe models as their PHY component. Parameter names that start with the prefix "mac." map directly to the selected EMANE MAC model. A list of the configuration parameters that can be specified through the XML scenario file is shown in Table 1.

3.2 Link Characterization

mLab performs link characterization by generating the appropriate EMANE *Pathloss* or *Location* events for each of the configured network emulation models (Fig. 4). For example, each time there is change in node position, mLab recalculates path loss values using mLab's theoretical or data-driven propagation model implementations and generates an EMANE *Pathloss* event to change the link conditions in the emulation environment. Instead, if we are leveraging completely on EMANE's capabilities, mLab generates Location events after changes in position to have EMANE dynamically compute path loss values using its own theoretical propagation models (this is the default behavior of the system).

3.3 Path Loss Feedback

As a consequence of doing link characterization using EMANE's own propagation models, mLab can no longer determine if any two given nodes can reach each other directly, and therefore it is unable to provide the required information to MView (mLab's visualizer) so that the link can be displayed depending on the connectivity between NEMs (Fig. 5). To solve this problem, we extended EMANE with a feedback mechanism that allows it to notify changes in link characteristics (i.e. path loss) back to mLab via UDP messages.

One of the challenges we encountered when doing the necessary changes to support this feature in EMANE was that, by design, path loss computation only occurs

Table 1 Subset of the most relevant configuration parameters that can be specified in a mLab scenario file

Name	Description	Default	80211abg	RFPipe
phy.bandwidth	Center frequency bandwidth in KHz.	20000	✓	✓
phy.antennagain	Antenna gain in dBi.	0	✓	✓
phy.systemnoisefigure	System noise figure in dB.	4.0	✓	✓
phy.pathlossmode	path loss mode of operation (path loss, freespace, 2ray).	freespace	✓	✓
phy.txpower	Transmit power in dBm.	15	✓	✓
phy.frequency	Transmit center frequency in KHz.	2417000	✓	✓
mac.enablepromiscuousmode	Determines if all packets received over the air will be sent up the stack to the transport layer.	off	✓	✓
mac.rtsthresh	Minimum packet size in bytes required to trigger RTS/CTS for unicast transmissions.	0	✓	
mac.datarate	Datarate/burstrate (Kbps) of the waveform being emulated.	1000		✓
mac.delay	Delay (usec) that is to be included in the transmission delay.	0		✓
mac.jitter	Jitter (usec) to be included to the transmission delay.	0		✓

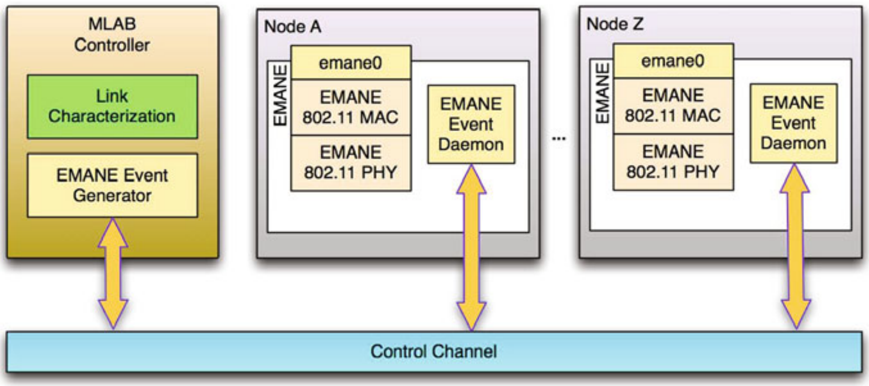
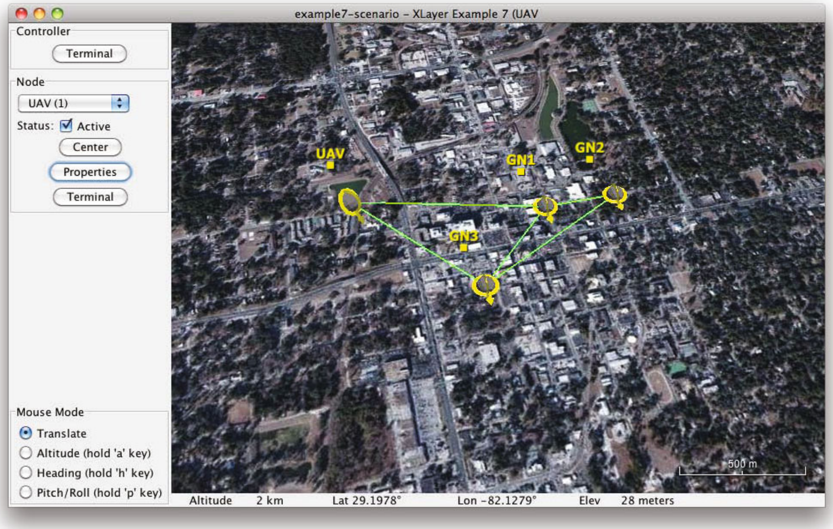


Fig. 4 mLab - EMANE Integration


example7-scenario – XLayer Example 7 (UAV

Node: UAV (1)

Status: ☑ Active

Mouse Mode

● Translate
○ Altitude (hold 'a' key)
○ Heading (hold 'h' key)
○ Pitch/Roll (hold 'p' key)

Altitude 2 km Lat 29.1978° Lon –82.1279° Elev 28 meters

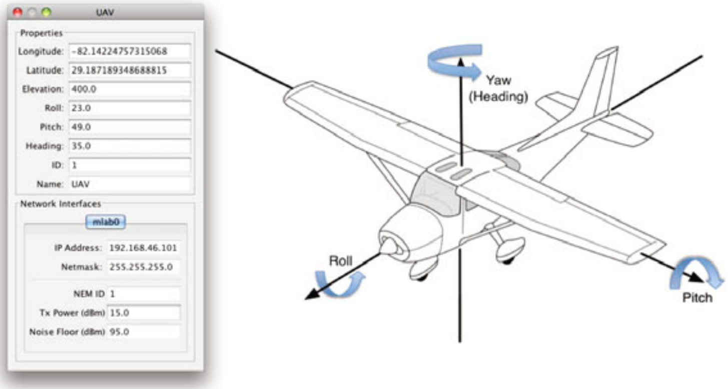
Fig. 5 Lines represent links and indicate connectivity between nodes (NEMs)

upon reception of data packets, which means that EMANE cannot provide any path loss information when there are no transmissions. To solve this problem, we had to modify EMANE so that path loss is also computed when *Location* events are received and handled by the Universal PHY Layer. In this case, EMANE sends a message to mLab containing the new values for path loss, which then estimates the connectivity between two nodes and passes the information to MView so that links can be properly visualized.

3.4 Transmission Power Control

An additional modification to EMANE was implemented to allow mLab to dynamically manipulate the transmitter power, which renders useful when doing transmitter power-based topology control, for example. In EMANE, parameter values are specified in the XML configuration files and cannot be changed dynamically, however, using EMANE's event mechanism we defined the TxPower event and extended the Universal PHY Layer to handle it. The event contains the NEM ID and the desired transmitter power level (dBm) for the corresponding network interface, specified by the user in the Node Properties dialog of MView (Fig. 6). Upon reception of the TxPower event, the Universal PHY Layer automatically adjusts the TX power setting to the desired value. It is worth noting that the same approach could be utilized to change other configuration parameters if the necessary events are made available.

The main advantage of this approach is that it is clean and fits naturally into EMANE's architecture; however, one of the disadvantages is that user-specified settings (e.g. transmitter power) can be overridden by the model's MAC layer through



UAV

Properties

Longitude: -82.14224757315068

Latitude: 29.187189348688815

Elevation: 400.0

Roll: 23.0

Pitch: 49.0

Heading: 35.0

ID: 1

Name: UAV

Network Interfaces

mlab0

IP Address: 192.168.46.101

Netmask: 255.255.255.0

NEM ID 1

Tx Power (dBm) 15.0

Noise Floor (dBm) 95.0

Fig. 6 The Node Properties dialog in MView

the MAC-PHY Control Messaging API, which provides the MAC component with the ability to override default PHY configuration parameters. If this is the case, then the TX power level specified by mLab using the TxPower event will have no effect.

One possible solution would be to extend the MAC layer to also handle the TxPower event, however, it will require modifying every possible model that provides its own MAC implementation. Therefore, we assume that when transmitter power control is required, the MAC of the selected model has not been configured to override PHY's configuration parameters, in particular the transmitter power.

4 Statistical Link Modeling

As part of mLab, we have developed and validated a statistical data driven model extracted from PENGWUN datasets to predict the link characteristics and network dynamics of an airborne network. In most cases, theoretical path loss models only consider distance between nodes for estimating the received signal strength indicator (RSSI) of the wireless link. However, in airborne communications signal is also attenuated when the ground and aircraft antennas lose line of sight when the aircraft is banking. Hence, we have used the PENGWUN datasets to create models for estimating the RSSI of wireless links in airborne networks based on distinct parameters.

The PENGWUN datasets consist of measurements of different platform, network and communication parameters such as GPS positions of ground nodes and aircrafts, heading, elevation and bank angle of the aircraft, RSSI and packet loss. These measurements are taken during actual field tests and saved to log files that are later imported into a database for further processing. This post-processing includes the automatic consolidation and synchronization of logged data from source and target nodes that is used in the generation of the statistical data driven models.

In the PENGWUN datasets, the variables that mostly affect the RSSI are the distance, because the signal gets attenuated with the distance, and the roll or banking of the airplane, because the ground antenna and the airplane antenna temporarily lose line of sight when the airplane is banking. Our models only consider these two dimensions when estimating the RSSI. Fig. 7 shows the scatter plots of the distance vs. RSSI and Figure 8 shows the relationship between banking angle vs. RSSI for one of the flights. The distance seems to be a better predictor for the RSSI than the angle, according to these graphs, but notice that at certain points (approx. 200m and 400m) the RSSI varies considerably. If at these points we look at the angle (see Fig. 9), this high variability can be explained. The variability for these distances occurs when the banking angle changes from 0° to 40°.

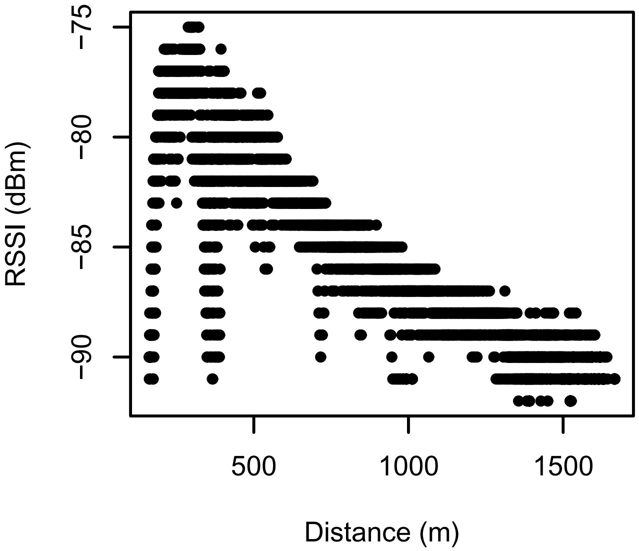


Fig. 7 Scatter plots of Distance vs. RSSI

Using the experimental data, two types of models are proposed for estimating the RSSI. The first type of model being proposed is an empirical model constructed using non-parametric estimation. The main advantage of this approach is that we can introduce as many dimensions as necessary without knowing the real distribution of the data. One of the disadvantages, however, is that the convergence to true values takes longer and it is proportional to the number of dimensions used for the estimation.

The second type of model being proposed is a parametric model based on a theoretical propagation model. The propagation model that was used is known as the log-normal shadowing model [15]. For this model two parameters need to be estimated from the data: the path loss exponent and the variance. The main advantage

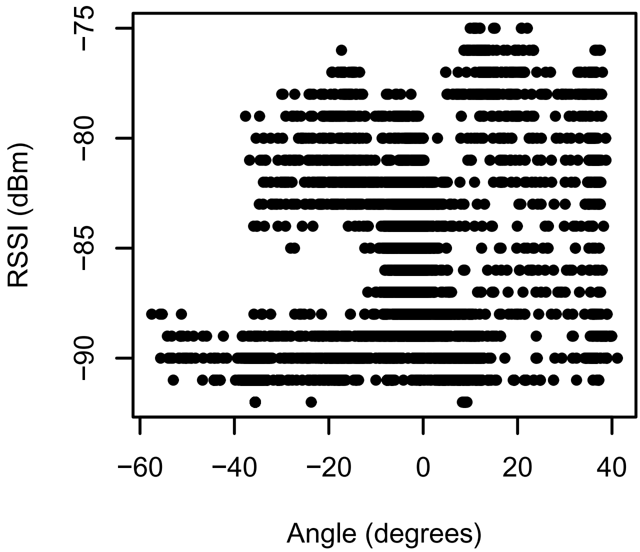


Fig. 8 Scatter plots of Distance vs. RSSI

of this model is that convergence is much faster. One disadvantage with this type of model is that we can only use the dimensions used by the model for the estimation, which in this case is only the distance. A second disadvantage with this type of model is that we are assuming that the distribution of the experimental data is equal to the distribution proposed by the model.

4.1 Empirical Model

For the empirical model, two predictor variables are used to predict the RSSI: the distance and the banking angle of the airplane, also known as the roll. The Cartesian plane formed by these two dimensions is divided in a grid with squares of 10m increments in the distance dimension and 1 degree increments in the banking angle dimension. The distance dimension ranges approximately from 160m to 1700m. The banking angle dimension ranges approximately from -60 degrees to 40 degrees.

For each square of the grid, the RSSI is modeled using the empirical distribution function, which is commonly used as a non-parametric method for estimating the cumulative distribution function (cdf) of an unknown distribution. Let $r_1, r_2, \dots r_n$ be RSSI data points sampled from the common unknown distribution function. The cdf of the empirical distribution is defined by:

$$F(r) = \frac{1}{n}\sum_{i=1}^{n} I(r_i \leq r), \tag{1}$$

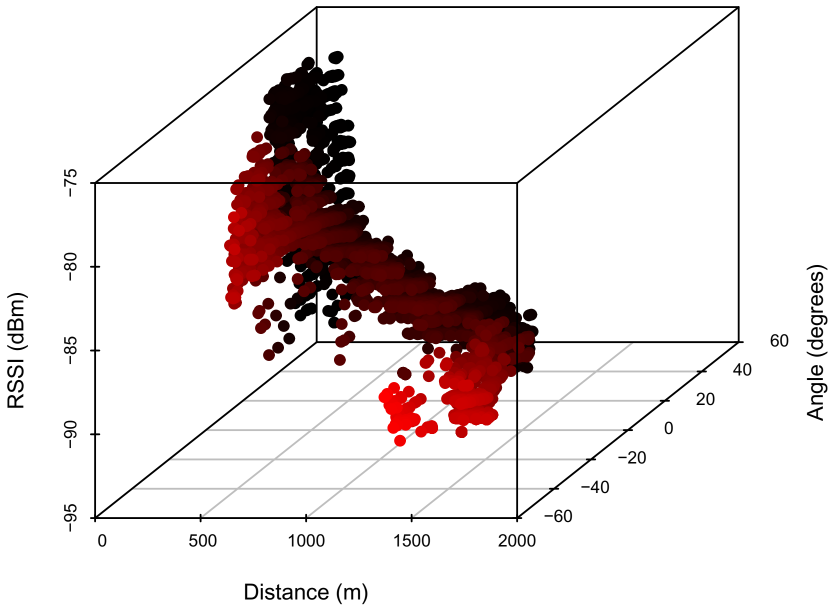


Fig. 9 Scatter plot of Distance and Angle vs. RSSI

where r is the RSSI for which the cumulative probability is being computed and $I(r_i \leq r)$ is the indicator function, defined by:

$$I(r_i \leq r) = \begin{cases} 1 & \text{if } r_i \leq r \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Having modeled the RSSI for each square in this way, we can now generate pseudo-random numbers drawn from this distribution by generating a $p \sim U(0,1)$, and then computing $F^{-1}(p)$. So, to estimate the RSSI for a given distance and banking angle, we find the corresponding square, and then use the empirical cdf for that square to randomly generate the RSSI value.

4.2 Theoretical Model

The theoretical model used is known as the log-normal shadowing model [15], which is the same probabilistic model used by NS-2. This model assumes that the average received signal power decreases logarithmically with distance, and that the path loss is randomly distributed log-normally (normal in dB) about that mean. The equation describing the path loss at a given distance is:

$$PL(d) = \overline{PL}(d_0) + 10 \cdot \beta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma, \tag{3}$$

where $\overline{PL}(d_0)$ is the average path loss at the close-in reference distance which is based on measurements close to the transmitter or on a free space assumption from the transmitter at distance d_0, β is the path loss exponent which indicates the rate at which the path loss increases with distance, and X_σ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB).

Using the free space assumption, the average path loss at the close-in reference distance can be estimated using the following equation:

$$PL(d) = -20 \cdot \log\left(\frac{\lambda}{4\pi d}\right), \tag{4}$$

where λ is the wave length of the carrier signal.

To be able to use the log-normal shadowing model, we need to estimate two parameters from the experimental data: the path loss exponent, and the standard deviation of the random variable. These two parameters can be estimated by doing a linear regression of the path loss using as predictor the logarithm of the distance. The path loss in the experimental data for any given point can be computed using the following formula:

$$PL = P_t - P_r, \tag{5}$$

where *Pt* is the transmitted power and *Pr* is the received power, both in *dBm*. The slope of the regression line can be used as an estimate of the path loss exponent by dividing the slope by 10, and the variance obtained from the regression can be used as an estimate of the variance of the random variable. For example, using the data from the same flight shown in Figs. 7 and 8, the path loss is 0.127 and the deviation is 0.25. Fig. 10 illustrates the relationship between RSSI and distance, for the values generated using a log normal shadowing model with the same parameters.

4.3 Model Validation

Validating the models means making sure that data generated by the models resembles the distribution of the experimental data. Using a goodness-of-fit test would only be appropriate for validating the theoretical model, because the empirical model is non-parametric.

It would be ideal to use a validation method that would allow us to compare which model fits better. With this purpose in mind, we used the following validation process. Let $\{(d_1,r_1,p_1),(d_2,r_2,p_2)\ldots(d_n,r_n,p_n)\}$ be the testing data, composed of distance (d_i), roll angle (r_i) and RSSI (p_i) tuples. For each of the models to validate, we will generate points $\{(d_1,r_1,\hat{p}_1),(d_2,r_2,\hat{p}_2)\ldots(d_n,r_n,\hat{p}_n)\}$, where $\hat{p}_i$ is the estimated RSSI from the model being validated.

Dividing the testing and estimated data in the same fashion as it is divided in the empirical model (a grid divided by distance and angle), we can compute the average for each of the squares in the grid and then compute the relative error between the average of the testing data and the average for the estimated data for the same square in the grid.

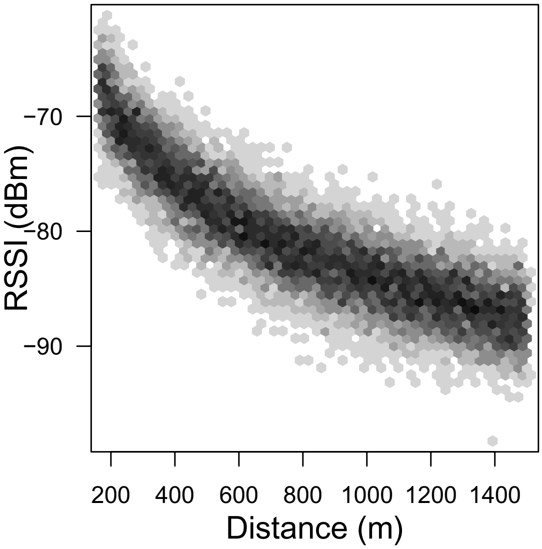


Fig. 10 Scatter plot of Distance vs. RSSI for RSSI points generated using the log-normal shadowing model

The errors for each of the squares in the grid can later be combined using a simple average, providing a single estimate of the error for the estimated model acquired for a given test set. If the models are then tested using multiple test sets, we can get a better estimate of the error, by averaging the error obtained for each of the test sets.

Ideally, the best way to test the models would be to use data from a different flight, but given that different flights may take place under different conditions, the estimates of the error for each of the models might be misleading. On the other hand, using for evaluation the same data that was used for estimating the models would not be appropriate as it would lead to over-fitting. A better way to empirically validate the models is to divide the experimental data in a training set and a testing set, using *k*-fold cross-validation [18] to estimate the accuracy of the model.

We used test flights from the PENGWUN datasets to validate and compare the accuracy of both empirical and theoretical models when predicting the RSSI of signal transmitted by an airborne node and measured from a node in the ground. We selected flights for which we have information about the characteristics of the transmitter, in particular the frequency and transmitter power.

Using the specified transmitter power and transmitting frequency, and using the measured RSSI at each data point, we computed the path loss exponent (PL) needed for the shadowing propagation model. With the calculated PL value in hand, we were now able to compute the theoretical estimated value of RSSI between the nodes of the dataset.

We estimated the accuracy of the theoretical model using k-fold cross-validation with 90% of the data used for training the model and the remaining 10% for validation. Likewise, we used the same data to compute and feed the empirical model with the distance and antenna angles between sender and receiver, and proceeded to perform k-fold cross validation using the same data partitions.

Table 2 shows the results of the validation of both the theoretical and empirical models. The results show that in all cases, the empirical model was able to successfully predict the RSSI of each link with an error rate that is a fraction of the error rate of the theoretical model. These results show that despite its disadvantages, the empirical model provides better estimates of the RSSI for airborne wireless links than the theoretical log-normal shadowing propagation model.

Table 2 Relative error rates between estimated and observed RSSI averages.

Flight No.	Source Node	Target Node	Empirical Model	Theoretical Model
			Error	Error
1	UAV	GN1	9%	57%
1	UAV	GN2	18%	47%
2	UAV	GN1	7%	53%
2	UAV	GN2	17%	48%
3	UAV	GN1	11%	56%
3	UAV	GN2	16%	57%

5 Conclusion

In this chapter we propose a hybrid emulation approach for airborne network environments, and describe the designed and implementation of the mLab-Pengwun emulation framework. mLab-Pengwun integrates an experimental flight test database with EMANE [13] to provide a high-fidelity link emulation capability for known flight patterns.

The proposed hybrid approach for airborne emulation environments allows for the design, test and evaluation of airborne protocols in laboratory, under comparable conditions expected for field tests. The separation of external link effects from the protocol design allows for an evaluation of the protocol properties under different realistic link conditions, without the need for repeated flight tests. Furthermore, it allows for field-validation tests of the proposed protocols, with the proper isolation of environmental and external effects that are generally not considered in theoretical link models.

The use of both theoretical propagation models and statistical data-driven models derived from experimental flight data makes mLab a well-fit emulation environment for airborne wireless networks, where multiple degrees of freedom and the complexity of the nodes make it very difficult to create reliable theoretical models that would suffice for the emulation. While there are multiple approaches in the literature for data-driven modeling, in this work we have chosen a simple parametric, and a non-parametric data model for links, with good results.

The proposed bi-dimensional statistical data-driven propagation model has shown to provide better estimates for RSSI values of airborne links than the theoretical propagation models that were considered for comparison in this work. The selection of parameters used for data-driven models (both parametric and non-parametric) used in this work were based on the flight experiments available in PENGWUN. While, based on our limited results, they cannot be claimed to be the best models in general, they have given us the best accuracy in our experimental tests.

Acknowledgements. This work was developed in part under sponsorship of the U.S. Air Force Research Laboratory, under contract number FA8750-10-2-0145.

References

1. The ns-3 network simulator (2009), `http://www.nsnam.org`
2. Ahrenholz, J., Danilov, C., Henderson, T.R., Kim, J.H.: Core: A real-time network emulator. In: IEEE Military Communications Conference, pp. 1–7 (2008)
3. Bajaj, L., Takai, M., Ahuja, R., Tang, K., Bagrodia, R., Gerla, M.: Glomosim: A scalable network simulation environment. University of California, Los Angeles (UCLA). Computer Science Department Technical Report 990027, 213 (1999)
4. Chao, W., Macker, J., Weston, J.: NRL mobile network emulator (2003)
5. Erceg, V., Greenstein, L.J., Tjandra, S.Y., Parkoff, S.R., Gupta, A., Kulic, B., Julius, A.A., Bianchi, R.: An empirically based path loss model for wireless channels in suburban environments. IEEE Journal on Selected Areas in Communications 17(7), 1205–1211 (1999)
6. Hortelano, J., Cano, J.C., Calafate, C.T., Manzoni, P.: Testing applications in manet environments through emulation. EURASIP Journal on Wireless Communications and Networking, 47 (2009)
7. Ivanic, N., Rivera, B., Adamson, B.: Mobile ad hoc network emulation environment. In: IEEE Military Communications Conference, MILCOM 2009, pp. 1–6 (2009)
8. Jiang, W., Zhang, C.: A portable real-time emulator for testing multi-radio manets. In: 20th International Parallel and Distributed Processing Symposium, IPDPS 2006, pp. 7–10. IEEE (2006)
9. Kiess, W., Mauve, M.: A survey on real-world implementations of mobile ad-hoc networks. Ad Hoc Networks 5(3), 324–339 (2007)
10. Konishi, K., Maeda, K., Sato, K., Yamasaki, A., Yamaguchi, H., Yasumoto, K., Higashino, T.: MobiREAL simulator-evaluating MANET applications in real environments. In: 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 499–502. IEEE (2005)
11. Kotz, D., Newport, C., Gray, R., Liu, J., Yuan, Y., Elliott, C.: Experimental evaluation of wireless simulation assumptions. In: Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 78–82. ACM (2004)
12. Kurkowski, S., Camp, T., Colagrosso, M.: Manet simulation studies: the incredibles. SIGMOBILE Mob. Comput. Commun. Rev. 9, 50–61 (2005)
13. Labs, C.: Emane user training workbook: 0.6.2. Tech. rep., CenGen Inc.,, Columbia, Maryland (2010)
14. Pawlikowski, K., Jeong, H.D.J., Lee, J.S.R.: On credibility of simulation studies of telecommunication networks. IEEE Communications Magazine 40(1), 132–139 (2002)

15. Rappaport, T.S.: Wireless communications: principles and practice. Prentice Hall PTR, New Jersey (1996)
16. Setiwan, G., Iskander, S., Chen, Q., Kanhere, S., Lan, K.: The effect of radio models on vehicular network simulations. In: Proceedings of the 14th World Congress on Intelligent Transport Systems (2007)
17. Shrivastava, L., Jain, R.: Performance analysis of path loss propagation models in wireless ad-hoc network. International Journal of Research and Reviews in Computer Science 2(3) (2011)
18. Wasserman, L.: All of statistics. Springer, New York (2004)
19. Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.: Impact of radio irregularity on wireless sensor networks. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 125–138. ACM (2004)